|  |  |
| --- | --- |
| MODELO INCREMENTAL | MODELO EN CASCADA |
| Combina elementos del MLS con la construcción de prototipo. Se divide en 4 partes: Análisis, Diseño, Código y Prueba. Para la producción del Software, se usa el principio de trabajo en cadena, Con esto se mantiene al cliente en contacto con los resultados en cada incremento a fin de que el software se adapte mejor a sus necesidades reales hasta que se elabore el producto completo.  Se evitan proyectos largos y se entrega algo de valor a los usuarios con cierta frecuencia.  - El usuario se involucra más.  - Difícil de evaluar el coste total.  - Difícil de aplicar a los sistemas transaccionales que tienden a ser integrados y a operar como un todo.  - Requiere gestores experimentados.  - Los errores en los requisitos se detectan tarde.  - El resultado puede ser muy positivo. | Es el enfoque metodológico que ordena rigurosamente las etapas del ciclo de vida del software, de tal forma que el inicio de cada etapa debe esperar a la finalización de la inmediatamente anterior. Un ejemplo de una metodología de desarrollo en cascada es:  1.- Análisis de requisitos 2.- Diseño del Sistema 3.- Diseño del Programa 4.- Codificación 5.- Pruebas 6.- Implantación  7.- Mantenimiento  Así cualquier error de diseño detectado conduce necesariamente al rediseño y nueva programación del código afectado, aumentando los costos del desarrollo. |

|  |  |
| --- | --- |
| MODELO EN V | MODELO SASHIMI |
| Es una representación grafica del desarrollo de un sistema, resume las principales medidas. Es una variación del modelo en cascada que muestra cómo se relacionan las actividades de prueba con el análisis y el diseño significa verificación y validación. Sus partes son: inicio, especificaciones, diseño prediminal, diseño en detalle, programación, prueba de unidad, integración, calificación, fin.  Busca regular el proceso de desarrollo del Software, minimización de riesgos del proyecto mejorándolo y garantizándolo. Su objetivo es definir las fases para validar el desarrollo de aplicación, para que el software cumpla los requisitos de aplicación. | Se permite un solapamiento entre fases, sin tener terminado del todo el diseño se comienza a implementar. Una ventaja de este modelo es que no necesita generar tanta documentación como el ciclo de vida en cascada. Es una variación del ciclo de vida en cascada puro, en el cual las diferentes etapas pueden ser solapadas, permitiendo así aumentar la eficiencia mediante la retroalimentación entre las etapas. Se obtiene una ganancia de calidad en el producto final, no hace necesario una documentación detallada para cada etapa, ya que por el mismo hecho de que estas se solapan, comparten partes de la documentación. Un problema es que existe la dificultad de identificar el inicio y el fin de cada etapa, en caso de presentarse problemas de comunicación estos van a generar inconsistencias. |